Wine Quality Prediction

Evan Gousha, egousha@bellarmine.edu

Gabe Migliore, gmigliore@bellarmine.edu

ABSTRACT

Wine taste is determined by several variables, and each is important towards overall quality. Using four types of regression analysis, we were able to determine the most important variables. We split the wine into training, testing, and validation sets. 99% in training or testing, and 1% in validation. Using Python we imported the datasheet using numpy and graphed the data using seaborn. Seaborn was important because it allowed us to see oddities in the data, none were present. Using mean squared error values we determined that Support Vector Regression was the most accurate. Problems encountered was the fact that quality had a relatively large range, but all the independent variables were rather clustered together, so it is hard to supervise the machine and see if the variance is accurate. After concluding the tests, we found that perhaps objective data cannot be used to predict wine quality but instead subjective opinion is more important.

INTRODUCTION

Wine is created through a process of harvesting, crushing, fermentation, clarification, and finally aging and bottling. Over time competitions have been made to find the best combination of these in order to create the best tasting wine. Competitions are held and overall quality of the wine is determined by taste, texture, depth, and overall consistency. Our data compares different tasting wines and their general chemical baselines to determine what is important in a good quality wine.

BACKGROUND

DATA SET DESCRIPTION

Definitions for variables used in data:

Fixed Acidity: most acids involved with wine that are fixed and nonvolatile.

Volatile Acidity: the amount of acetic acid in wine, in high levels can lead to an unpleasant vinegar taste.

Citric Acid: found in small quantities, citric acid adds ‘freshness’ and flavor.

Residual Sugar: The amount of sugar remaining after fermentation stops. Chlorides: the amount of salt.

Free Sulfur Dioxide: the free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion.

Total Sulfur Dioxide: amount of free and bound forms of S02.

Density: the density of water.

pH: describes the acidic or alkalinity of a wine from 0 (acidic) to 14(alkaline).

Sulfates: a wine additive that contributes to SO2 levels.

Alcohol: the percent of alcohol content.

MACHINE LEARNING MODEL

Four regression models used:

Multiple Linear Regression: Uses statistics to assign importance to different variables and predict outcomes.

Support Vector Regression: a supervised learning algorithm which can be used for regression as well as classification problems.

Decision Tree Regression: a supervised learning algorithm which can be used for solving both classification and regression problems.

Random Forest Regression: one of the most powerful supervised learning algorithms which is capable of performing regression as well as classification tasks.

EXPLORATORY ANALYSIS

Our data set is a relatively large one, originally having 12 columns with 1143 non null values in each column, however when preparing for the models, we eliminated one of these columns leaving us with only 11. When looking at the histograms of the variables, despite some positive skewness on some of the variables, we felt it wasn't anything odd enough to really alter our model or skew our results.

DATA TYPES

VARIABLE NAME	**VARIABLE TYPE**
Fixed Acidity	Numerical
Volatile Acidity	Numerical
Citric Acid	Numerical
Residual Sugar	Numerical
Chlorides	Numerical
Free Sulfur Dioxide	Numerical

Total Sulfur Dioxide	Numerical
Density	Numerical
pH	Numerical
Sulfates	Numerical
Alcohol	Numerical
Quality	Numerical
Id	Categorical

METHODS

DATA PREPARATION

In order to prepare the data from modeling, we had to first analyze the different variables that made up the data set, and ask ourselves which we thought were genuinely needed to predict the quality. After analyzing them, we noted that almost all were required to determine the quality because they were all important factors to the different characteristics of wine. However, we did not need the 'Id' column because that was just there to indicate the different wines and didn't affect quality. Besides the simple column drop, the data was essentially already ready to model, none of the columns had empty cells, so filling null values was unnecessary.

EXPERIMENTAL DESIGN

In order to capture the most accurate recordings possible, we ran each of the four tests in different ways. We first noted that the two tools provided different predictions purely based off of their software, as well as the different splits. Considering our data set was so large, having a large validation set was very impractical, so we landed on having around a 10% validation set with 10 randomly selected wines to predict the quality of. The rest of the 99% was split into testing and training, with us using two different splits for each software to maximize results.

EXPERIMENTAL PARAMETER

EXPERIMENT NUMBER	**PARAMETERS**
1	All 4 tests using Python's regression, 80/19/1 split for train, test, and validate
2	All 4 tests using Python's regression, 70/29/1 split for train, test, and validate
3	All 4 tests using R's regression, 80/19/1 split for train, test, and validate
4	All 4 tests using R's regression, 70/19/1 split for train, test, and validate

TOOLS USED

For this project, we only used python's different regression models, along with R's different regression models. Within Python, when starting we used Seaborn in order to graph histograms of all the variables in order to see any oddities, but it wasn't really needed as all the variables seemed to work easily into the model.

RESULTS

MEAN SQUARE ERROR AND R-SQUARED CALCULATION

PROGRAM	**MODEL**	**MEAN SQUARE ERROR**	**R-SQUARED**
Python	Multiple Linear Regression 80/19/1	0.60520	0.40142
Python	Support Vector Regression 80/19/1	0.60695	0.39797
Python	Decision Tree Regression 80/19/1	0.60222	0.38988
Python	Random Forest Regression 80/19/1	0.59368	0.42400
Python	Multiple Linear Regression 70/29/1	0.60614	0.381772
Python	Support Vector Regression 70/29/1	0.58630	0.421594
Python	Decision Tree Regression 70/29/1	0.78961	-0.049109 ???
Python	Random Forest Regression 70/29/1	0.60222	0.38975
R	Multiple Linear Regression 80/19/1	0.641045	0.3577
R	Support Vector Regression 80/19/1	0.5971	0.4271738
R	Decision Tree Regression 80/19/1	0.69189	0.3273603
R	Random Forest Regression 80/19/1	0.7241	0.1677275
R	Multiple Linear Regression 70/29/1	0.63952	0.3822863
R	Support Vector Regression 70/29/1	0.6000	0.423801
R	Decision Tree Regression 70/29/1	0.67198	0.3195981
R	Random Forest Regression 70/29/1	0.74621	0.1539515

DISCUSSION OF RESULTS

Because of the number of times we ran the tests, we came to somewhat expect which models were going to be the most accurate. If we judge the results entirely off of the mean squared error values, it would appear that the Python

Support Vector Regression model with the 70/29/1 split provided the most accurate predictions, with the Python model's generally having more accurate results with the exception of the Decision Tree which we had problems with, explaining why it has such a high error rate. We believe the flexibility built into the SVR model allowed it to provide the most accurate results.

PROBLEMS ENCOUNTERED

The only real problem we encountered was using the decision tree model, for both Python and R. In Python, the answers were often odd and we occasionally encountered some errors, and often required running the test over and over to get answers that made sense. Within R, the answers usually came out fine, but we had some repeats which leaves us to wonder about their accuracy. The rest of the models worked out pleasantly, with only minor inconveniences every now and then.

LIMITATIONS OF IMPLEMENTATION

Clearly our models had their limitations as each had a high mean squared error considering the range of values that our quality column. We think that there is the biggest limitation, with the data being such a small cluster of values that it's hard to tell whether the models correctly guessed the quality or if it could just pick a random number within a small range and still be accurate. Also with the low r squared values, it's clear that the variables didn't have much relation on the quality, so the association is minimal.

IMPROVEMENT/FUTURE WORK

To improve this experiment, we think that possibly narrowing down the variables could possibly provide more consistent results, and see the individual relationships between the variables and the quality. However this would be hard considering each variable makes up the wine, so narrowing the variables would lose the bigger picture. Generally, a different dataset might be a better option when wanting to apply these models.

CONCLUSION

In conclusion, our problem was to try and predict the quality of wine based on a list of variables that make up the contents of wine. To do this, we used regression models which is a statistical method used to determine the relationship between independent variables and a particular dependent variable. In our case, our dependent variable was the quality of the wine. We used four types of regression tests, using two different testing-training-validation splits as well as two different programs, for a total of sixteen regression models. The model that proved the most accurate results was Python's 70/29/1 SVR model which had the lowest mean squared error of about 0.586. However despite this being the most accurate, does it really give us helpful insight? We would lean toward the answer of no. The models had relatively high error rates when it comes to our predictions and seemed unreliable for the most part. This begs the question of whether the quality of wine really is affected by its quantitative inputs, or whether the subjective nature of quality overrules all independent variables.